Using noun phrases and tf-idf for plagiarized document retrieval

Notebook for PAN at CLEF 2014

Victoria Elizalde

kivielizalde@gmail.com

Abstract This paper describes an approach submitted to the 2014 PAN competition for the source retrieval sub-task[7]. Both independent term and phrasal queries are generated, using either term frequency-inverse document frequency or noun phrases to select the terms.

1 Introduction

With the advent of the Internet, plagiarism has become a widespread problem, affecting Universities and other academic institutions. While earlier detection methods included searching for sources in a local database, today a realistic approach must also look for sources in the World Wide Web. This is done by inspecting the suspected document and generating queries which are then posed to one or more search engines, such as Google or Yahoo!. This task is known as Source Retrieval.

Within the PAN competition[5], two academic search engines are used: Indri[10] and ChatNoir[8]. These engines search the ClueWeb09 corpus[1], which consists of about 1 billion web pages collected from the World Wide Web in 2009. The suspicious documents come from a subset of the PAN-PC-10 corpus[9].

2 Source retrieval

The approach presented to the competition this year is based in the 2013 submission[3], with some modifications. The software was developed using Python and the Natural Language Toolkit [2].

In order to shorten the software runtime, each document is processed in a different thread. Up to 6 different threads are executed at the same time, since this task depends more on results from search engines than on CPU time. The queries are sent in three batches for each document (one per strategy used in the query generation phase) to optimize network usage and avoid congestion in the search engines.

[1] http://www.lemurproject.org/clueweb09.php/

2.1 Query generation

This year's submission includes queries generated using term frequency-inverse document frequency (tf-idf) coefficient and Noun Phrase detection. All the queries generated were submitted to the Indri search engine.

Exactly as in last year's submission, the text was divided in 50 line chunks, non alphabetical characters and stopwords were removed. Lemmatization was applied using the WordNet lemmatizer[4] and words were ranked by their tf-idf coefficient. The list of frequency words used was generated using the Brown Corpus[6]. Finally, a query with the top 10 ranked words was generated for each chunk.

$$t = \{10\ terms\ with\ highest\ tf - idf\}$$

$$t_i = \{i^{th} term \in t\}$$

Some modifications were introduced this year in the query formation. The first term (the one with the higher tf-idf) is required to be present in the results, using Indri's #filreq operator. The last 5 terms (the ones with lowest tf-idf) are combined with the #or operator, to allow for more flexibility in the query.

$$\#filreq\ (t_0\ \#combine(t_0\ t_1\ t_2\ t_3\ t_4\ \#or(t_5\ t_6\ t_7\ t_8\ t_9)))$$

In addition to this, the 10 selected terms are used to generate phrasal queries. From each 50 line chunk, the first 8-gram with at least 3 words of intersection with the tf-idf terms is extracted and sent to the search engine.

A keyphrase extractor by Baker and Cornacchia[1], with slight modifications, was used exactly as in last year algorithm. Noun phrases are clustered according to the nouns they contain, and then the largest 20 clusters are selected. For example, in the phrase "the Church of Ireland", the phrase would count both towards the cluster "Church" and the cluster "Ireland". Phrases are then ranked by multiplying the number of words in the phrase by the number of phrases in the cluster. The 15 most rated phrases are then posed to the Indri query engine.

2.2 Download filtering

The queries are posed, limiting the results to 30. The first 10 results are selected and a 500 character snippet is requested and processed. The snippets in which more than 90% of the 4-grams are found in the original text are considered promising and downloaded. If no snippet is present, the result is not downloaded.

While the last downloaded result was reported to be a source by the Oracle, the snippet of the following result is downloaded and analyzed as mentioned before, until there aren't more results or a downloaded result is not a source.

3 Results and discussion

Comparing last and this year submissions is difficult since, at the moment of writing, the performance of last year's software in this year's corpus is not available. This is true

Table 1. PAN 2014 Source retrieval final results

Submission	Retrieval Performance			Workload		Time to 1st Detection		No	Runtime
	F_1	Precision	Recall	Queries	Downloads	Queries	Downloads	Detection	
elizalde14	0.34	0.40	0.39	54.5	33.2	16.4	3.9	7	04:02:00
kong14	0.12	0.08	0.48	83.5	207.1	85.7	24.9	6	24:03:31
prakash14	0.39	0.38	0.50	59.96	38.77	8.09	3.76	7	19:47:45
suchomel14	0.10	0.08	0.40	19.5	237.3	3.1	38.6	2	45:42:06
williams14	0.47	0.57	0.48	117.13	14.41	18.82	2.34	4	39:44:11
zubarev14	0.45	0.54	0.45	37.03	18.61	5.4	2.25	3	40:42:18

especially for recall, which could vary considerably depending on the plagiarism cases present in the corpus. Nevertheless, some considerations can be made.

Precision has clearly improved, (going from 12% last year to 40% this year) product of the different changes made to the filtering algorithm. This includes longer snippets, not downloading results without a snippet and analyzing n-grams instead of looking at individual words. This modifications also decreased considerably the number of downloads per suspicious document (from 107.22 to 33.2) and the downloads until 1^{st} detection (15.28 to 3.9). This suggests that comparing word n-grams of the snippet instead of individual words may yield better precision and lower downloads when retrieving documents in plagiarism detection.

On the other hand, the amount of queries per document has increased from 44.50 to 54.5 product of the changes made to the query generation algorithm.

Using multiple threads has proven useful: the run time of the submitted approach is the lowest by far. This simplifies testing since executing a run takes less time.

The main objective for future work is to improve recall, without compromising precision.

References

1. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases (2000)
2. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly, Beijing (2009), http://www.nltk.org/book
3. Elizalde, V.: Using statistic and semantic analysis to detect plagiarism - notebook for pan at clef 2013. In: Pamela Forner, Roberto Navigli, and Dan Tufis, editors. CLEF 2013 Evaluation Labs and Workshop - Working Notes Papers, 23-26 September, Valencia, Spain, 2013 (2013)
4. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
5. Ferro, N., Forner, P., MÃijller, H., Navigli, R., Paredes, R., Rosso, P., Stein, B., Tufis, D.: 4th international conference of the clef initiative (clef 13). ACM SIGIR Forum 47(2), 15–20 (2013)
6. Francis, W.N., Kucera, H.: Brown corpus manual. Tech. rep., Department of Linguistics, Brown University, Providence, Rhode Island, US (1979), http://icame.uib.no/brown/bcm.html
7. Potthast, M., Hagen, M., Gollub, T., Tippmann, M., Kiesel, J., Rosso, P., Stamatatos, E., Stein, B.: Overview of the 5th international competition on plagiarism detection. In: P., T.D.E..F. (ed.) Notebook Papers of CLEF 2013 LABs and Workshops (CLEF-2013) (2013)

8. Potthast, M., Hagen, M., Stein, B., Graßegger, J., Michel, M., Tippmann, M., Welsch, C.: ChatNoir: A Search Engine for the ClueWeb09 Corpus. In: Hersh, B., Callan, J., Maarek, Y., Sanderson, M. (eds.) 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 12). p. 1004. ACM (Aug 2012)
9. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. In: Huang, C.R., Jurafsky, D. (eds.) COLING (Posters). pp. 997–1005. Chinese Information Processing Society of China (2010)
10. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: a language-model based search engine for complex queries. Tech. rep., in Proceedings of the International Conference on Intelligent Analysis (2005)